

Appeal Brief  
Appl. No. 09/931,296  
Amended: February 13, 2006



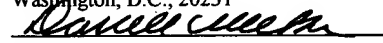
02-14-06

AF *FW*

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Appl. No.: 09/931,296 Confirmation No.: 3181  
Applicant: Dutta et al.  
Filed: August 16, 2001  
TC/A. U. 2171  
Examiner: Macin R. Filipczyk  
Docket No.: AUS920010531US1  
Customer No.: 46129  
Title: A METHOD AND SYSTEM FOR STORAGE, RETRIEVAL AND  
EXECUTION OF LEGACY SOFTWARE

Commissioner for Patents  
P. O. Box 1450  
Alexandria, Virginia 22313-1450

**"Express Mail" Mailing Label**  
Number **EQ 169318238** US  
Date of Deposit February 13, 2006  
I hereby certify that this paper or fee is being deposited  
with the United States Postal Services "Express Mail  
Post Office to Addressee" service under 37 CFR 1.10  
on the date indicated above and is addressed to the  
Commissioner of Patents and Trademarks,  
Washington, D.C., 20231  
  
Darcell Walker, Reg. No. 34,945

**TRANSMITTAL OF AMENDED APPELLANT'S BRIEF**

Applicant files the attached amended Appeal Brief in support of the Notice of Appeal filed by Applicant on August 1, 2005 in the above-identified application. Applicant files this amended appeal brief in response to a Notification of Non-compliant appeal brief mailed to Applicant on January 13, 2006. Therefore the filing of the brief is considered timely filed.

Respectfully submitted,

  
Darcell Walker

Reg. No. 34,945  
9301 Southwest Freeway, Suite 250  
Houston, Texas 77074  
713-772-1255  
February 13, 2006

Appeal Brief  
Appl. No. 09/931,296  
Amended: February 13, 2006

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

Appl. No.: 09/931,296

Confirmation No.: 3181

Applicant: Dutta et al.

Filed: August 16, 2001

TC/A. U. 2171

Examiner: Macin R. Filipczyk

Docket No.: AUS920010531US1

Customer No.: 46129

Title: A METHOD AND SYSTEM FOR STORAGE, RETRIEVAL AND  
EXECUTION OF LEGACY SOFTWARE



Commissioner for Patents  
P. O. Box 1450  
Alexandria, Virginia 22313-1450

**"Express Mail" Mailing Label**  
**Number: EO 169318238 US**

Date of Deposit February 13, 2006

I hereby certify that this paper or fee is being deposited  
with the United States Postal Services "Express Mail  
Post Office to Addressee" service under 37 CFR 1.10

on the date indicated above and is addressed to the  
Commissioner of Patents and Trademarks,  
Washington, D.C., 20231

  
Darcell Walker, Reg. No. 34,945

**APPELLANT'S BRIEF  
IN RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF  
(37 CFR 41.37)**

This brief is filed in triplicate in support of the previously filed Notice of Appeal, which was filed August 1, 2005, which appealed from the decision of the examiner dated March 1, 2005, rejecting claims 1-28.

Applicant has amended this brief to address the non-compliant issues of an office letter dated January 13, 2006. Applicant has included in the descriptions of the independent claims to include specific reference to the specification by page and line number and to the drawings the claimed functions in each independent claim.

**1. REAL PARTY IN INTEREST**

The real party in interest in this appeal is International Business Machines Corporation (IBM).

**2. RELATED APPEALS AND INTERFERENCES**

With respect to other appeals or interferences that will directly affect, or be directly affected by, or have a bearing on the Board's decision in the pending appeal, there are no such appeals or interferences.

**3. STATUS OF CLAIMS**

Claims 1-4, 6-11 and 13-28 are pending in this application; claims 1-4, 6-11 and 13-28 have been finally rejected; claims 1-4, 6-11 and 13-28 have been appealed. Claims 5 and 12 have been canceled. No claims have been allowed.

**4. STATUS OF AMENDMENTS**

Applicant has filed an amendment after final to address objections to the specification.

**5. SUMMARY OF THE CLAIMS**

Claim 1 describes a system for maintaining, accessing and executing legacy software programs stored in a central location. This system has a database storage location (10) (12a, 12b, 12c and 13) for storing software programs that have been created over time. Each created software program is a separate and independent program. A computing device (17) is connected to the database (10). The computing device (17) is capable of interacting with the database (10) for the purpose of retrieving and executing software programs stored in the database. An interface device enables communication between the computing device and a user. A computing network (22) provides the means for connecting the interface device (23) to the computing device to provide interaction between the computing device (17) and a user via the interface device (23).

Claim 10 describes a method for maintaining, accessing and executing legacy computer software programs stored in a central location. This method in this claim creates a database for storing multiple types of software programs (paragraph [0029], line 1). When this database receives a request from a user for a particular software program, the method accesses the database, submits the software program request to a database controller (51). The requested program is retrieved (paragraph [0035], line 5). Once retrieved, the next step is to make a determination whether to execute the retrieved software program (64). There can be occasions when it is not possible to execute a particular program at a particular time or for a particular request. The last step is to execute the retrieved software program (265) when the determination is to execute the retrieved software program.

Claim 16 describes a method for accessing and executing software computer programs stored in a central database location. This method accesses to computer software programs stored in a central location (10, Figures 2 and 3). Some of the software programs have been developed several years in the past. The method accesses the database (10) storing multiple types software programs, and retrieves from a controller (17) connected to the database a software programs options list (Figure 2) and an index of software program types and actual software programs stored in the database 10. A request (51) is then submitted to the software controller (17) containing a selected software program option and an identified software program desired for execution. The next step is to retrieve (paragraph [0039]) from the database an identified software program. Finally, the identified software program is executed (265) as desired by the requesting party.

Claim 28 describes a system for maintaining, accessing and executing computer software programs. This system includes a mechanism for storing computer (10) software programs, mechanisms (17) for accessing and executing the stored computer software programs and mechanisms (17) for transmitting and receiving messages over a computer network (22). The mechanism for storing computer software programs is a database that can have a collection of computer programs that span a chronological range from 1950's to the present. This system also has a computing device (23) connectable to a distributed computing network. This computing network provides the ability to have remote access to the database. Lastly, emulation and simulation programs stored in the database are used to assist in the execution of the software programs stored in the database.

## **6. ISSUES**

The issues on appeal are:

Whether claims 1-4, 6-11 and 13-28 under 35 U.S.C. § 102(e) are anticipated by Bentley et al (U.S. Patent 6,341,291).

## 7. ARGUMENTS

Was 35 U.S.C. § 102(e) properly applied in a rejection of claims 1-4, 6-11 and 13-28 as being anticipated by Bentley et al. (U.S. Patent 6,341,291)?

### Arguments in support of separate patentability

#### Background discussion of legacy software programs

Applicants' invention provides a method and system for creating a software storage, retrieval, and execution facility that charts the history and development of software programs. This storage facility would contain collection of legacy software programs that span time from the introduction of computers and software beginning in the 1950's to the explosion of various software products being developed today. The intent is for each software versions to be stored in an executable form.

A user desiring to execute a particular program can access that program via the database and execute the program according to the user's desires. The program accessed by the user may be 20 years old and no longer supported by the creators or suppliers. For example, a user may want to use a FORTRAN program for the 1970's. These programs or specific editions of the program are not readily or commercially available today. With the present invention, the user could retrieve the desired executable edition of the program and perform the desired activity with the program.

In the database, there may be several WINDOWS operating systems, such as WINDOWS 95, WINDOWS 98, WINDOWS 2000 and WINDOWS 2002. All of the programs are WINDOWS operating systems. However, each program is a distinct product. A user may want to use the WINDOWS 95 program. This program is not the current version or the most preferred. But a user may want to use this version. In the present invention, there is not executable connection between this program and any other program in the database.

#### Initial review of the teachings of Bentley et al.

Bentley describes a system used in product design and development. Bentley provides a system that enables multiple persons working on a project to update the

product information. These updates/modifications are recorded in order for everyone to know what changes have occurred, when they occurred and why they have occurred. This capability is desirable to enable all persons involved to be on the same page with regard to the current status of a project.

Bentley also provides a data management structure for engineering design data comprising a plurality of components. Each component has a unique identifier, a set of fields, each field having a data type and a data value, and a program, which interprets and modifies the fields, and, optionally, a list of other dependent components. At least some of the components represent respective elements in a file-based computerized editing system.

Bentley further provides a repository for storing a plurality of engineering models. Each engineering model comprises engineering design data and includes a specific plurality of components. Each component has a unique identifier, a set of fields, each field having a data type and a data value, a program that interprets and modifies the fields, and, optionally, a list of other dependent components. At least some of the components represent respective elements in a file-based computerized editing system.

#### Contrasting Bentley to the present invention

Figure 1 of Applicant's present invention can be used to provide a conceptual explanation of the functioning of the invention and can provide a distinct contrast to Bentley. As shown, the database 10 contains various types of software programs. The database has a collection of software programs that have been developed over a period of time. The collection of programs represents systems programs, utility programs and application programs. This collection can be as far as any initial record of computing. As suggested, there can be software programs developed as early as the 1950's up to the present. As technology changes and improvements are introduced, new software products are developed that usually replace existing software products. These replaced programs become antique or legacy programs. In addition, manufactures discontinue to provide technical support for these programs in an attempt to promote the purchase of the newer programs. However, the antique programs are still capable of executing as

designed. In Applicant's present invention, an antique program can be accessed and executed when desired. As shown in block 17, this system also has other components including software programs that assist in the execution of a legacy software program.

In the implementation of Applicant's invention, a user can access the legacy software database via a computing network. The location of the execution of the legacy program can vary. The execution can be local at the user machine or it can occur at a central location such as the block 17.

Bentley provides a data management structure for engineering design data comprising a plurality of components. Each component has a unique identifier, a set of fields, each field having a data type and a data value, and a program, which interprets and modifies the fields, and, optionally, a list of other dependent components. At least some of the components represent respective elements in a file-based computerized editing system. Bentley provides a repository for storing a plurality of engineering models. Each engineering model comprises engineering design data and includes a specific plurality of components. Each component has a unique identifier, a set of fields, each field having a data type and a data value, a program that interprets and modifies the fields, and, optionally, a list of other dependent components. At least some of the components represent respective elements in a file-based computerized editing system. (Col. 2, lines 12-29).

The examiner asserts that Bentley creates a database for storing multiple types of software programs. The examiner cites figure 10, items 2 and 3 and col. 2, lines 20-29. First item 2 (a ProjectBank CDB) is a single file or a controlled collection of files containing the components 10 that comprise the current version of the project. However, the ProjectBank CDB 2 can also be some other form of long-term storage facility such as a structured store or a relational database. (col. 15, lines 24-29). As mentioned, component 10 has within it a program, which interprets and modifies the fields, and, optionally, a list of other dependent components. These programs appear to facilitate the interpretation and modification of fields in the components. To the Applicants, these programs described in Bentley serve system purposes similar to the programs in block 17 of Applicants' invention, but the programs in block 10 of Applicants' invention.



The file or storage facility, which can comprise item 2, is not a legacy software program. The program with component 10 for interpreting and modifying files is not legacy software as described in Applicant's invention. Item 3 is a project history file (col.15, lines 30-31). As mentioned, col.2, lines 20-29 discuss a program within a component that interprets and modifies fields. These programs are not the type one would externally access to perform some task. It appears that these programs with internally to monitor/track changes.

Many of the cites in Bentley relied on by the examiner do not support the examiner's assertions. 1) The examiner rejects claims 2 and 12 stating that Bentley discloses multiple software programs, each software program capable of executing in a computing environment (col. 15, lines 29-32). That actual cites reads as follows: "The ProjectBank Server Program 1 also creates, opens, and operates on and maintains the Project History file 3, which contains a list of all changes to components 10 in the project." 2) Regarding claims 3 and 13, the examiner asserts Bentley discloses software programs are stored in software directories according to the type of software program. (col. 17, lines 63-66). The actual cites reads as follows: "Each history chapter 40 also contains a list 42 of "key components" to designate high-level components contained in the chapter 40. The purpose of the key component list 42 is to increase performance of history chapter 40 searches." 3) Regarding claims 4 and 15, Bentley discloses software directories include directories for operating system programs, application programs and utility programs (col. 18, lines 8-15). The actual cites reads as follows: "ProjectBank Client programs 4 may also save extensive change-description information with each history chapter 40 to document the engineering purpose of the change, or other information as may be relevant. This process does not require special support from the ProjectBank History file 3 since such change-description information can simply be held by a new component 10 that is created for every history chapter 40." Applicants submit that the above described assertions for which the examiner based rejections of the claims are not supported by the cites listed by the examiner. Applicants further submit that these cites are consistent with the objective of Bentley, but not of Applicants present invention.

Applicant submits that some of the features of Applicants' present invention are

also features in other computer implemented applications. However, Applicants submit that the creation of a legacy software database containing executable software programs in a somewhat library format is a novel concept. The ability to retrieve a software program that may be 20 to 25 years of and execute that program is has not been implemented. With the focus on future software developments, Applicants submit that Bentley does not anticipate this concept.

In view of the above, Applicants respectfully submit US Patent 6,341,291 (Bentley) does not anticipate Applicants' described invention. Contrary to the Examiner's statements that all elements of Applicants' claims are disclosed in the cited reference, the element of a database, as described in Applicants' invention, for storing software programs, developed over a period of time, each software program being stored as a separate and independent software program is not disclosed in the cited reference. Therefore the 35 U.S.C. § 102(b) rejection of the claims should be withdrawn.

## 9. CONCLUSION

In view of the above arguments, it is respectfully urged that the rejection of the claims should not be sustained.

Respectfully Submitted,



Darcell Walker  
Reg. No. 34,945  
9301 Southwest Freeway, Suite 250  
Houston, Texas 77074  
713-772-1255  
February 13, 2006

## **APPENDIX OF CLAIMS**

1. A system for maintaining, accessing and executing legacy computer software programs stored in a central location comprising:
  - a database for storing software programs, developed over a period of time, each software program being stored as a separate and independent software program;
  - a computing device connected to said database capable of interacting with said database for the purpose of retrieving and executing software programs stored in said database;
  - a interface device connected to computing device for interacting with said computing device; and
  - a computing network for connecting said interface device and said computing device.
2. The system as described in claim 1 further comprising in said database, multiple of software programs, each software program being capable of executing in a computing environment.
3. The system as described in claim 2 wherein said software programs are stored in software directories according to the type of software program.
4. The system as described in claim 3 wherein said software directories include directories for operating system programs, application programs and utility programs.
5. (Canceled)
6. The system as described in claim 1 wherein said interface device is a computer terminal.

7. The system as described in claim 1 wherein said computing device is a server.
8. The system as described in claim 7 wherein said server device comprises:
  - a central processing unit;
  - a memory;
  - emulator and simulator programs capable of creating computing environments;
  - and
  - software retrieval programs for accessing and retrieving software programs stored in said database.
9. The system as described in claim 1 further comprising multiple interface devices capable of interacting with a server simultaneously.
10. A method for maintaining, accessing and executing legacy computer software programs stored in a central location comprising:
  - creating a database containing multiple types of software programs;
  - accessing a database that stores multiple types of software programs, developed over a period of time each program capable of executing in a computing environment;
  - submitting a software program request to a database controller;
  - retrieving the software program identified in the software program request;
  - determining whether to execute the retrieved software program; and
  - executing the retrieved software program when the determination is to execute the retrieved software program.
11. The method as described in claim 10 further comprising after said retrieving step, the step of analyzing the different attributes of the retrieved software program.
12. (Canceled)

13. The method as described in claim 10 wherein said database creating step comprises:

- creating a directory for each software program type;
- placing each software program in the appropriate directory; and
- assigning an identifier to each program.

14. The method as described in claim 13 further comprising the steps of:

- creating specific collections of software programs by linking selected software programs together, said collection of programs becoming a software set.

15. The method as described in claim 10 further comprising after said accessing step, the steps of:

- retrieving a software programs index, said index containing directories of each type software in the database and a list of each software program in each directory; and
- identifying a specific software program in a software directory.

16. A method for accessing and executing software computer programs stored in a central database location, some of said software programs being versions developed over a period of time, said method comprising the steps of:

- accessing the database storing multiple types of software programs;
- retrieving from a controller connected to the database a software programs options list and a index of software program types and actual software programs;
- submitting a request to the software controller containing a selected software programs option and an identified software program;
- retrieving from the database an identified software program; and
- implementing the selected software option on the identified and retrieved software program.

17. The method as described in claim 16 wherein said implementation step comprises determining whether said selected program option is possible to implement and sending a decline message to the location to the submission when a selected option is not possible to implement on the identified and retrieved software program.

18. The method as described in claim 16 wherein said implementation step comprises creating a computing environment for executing the identified and selected software program.

19. The method as described in claim 18 wherein said computer environment creating step comprises determining the computer hardware and software necessary to implement the selected option on the retrieved software program.

20. The method as described in claim 19 wherein said implementation step occurs in the database controller.

21. The method as described in claim 19 wherein said implementation step occurs in a computing environment located at the software program request submission location.

22. A computer program product in a computer readable medium for accessing and executing software computer programs stored in a central database location comprising:

instructions for accessing the database storing multiple types of software programs;

instructions for retrieving from a controller connected to the database a software programs options list and a index of software types and software program;

instructions for submitting a request to the software controller containing a selected software programs option and an identified software program;

instructions for retrieving from the database an identified software program; and

instructions for implementing the selected software option on the identified and retrieved software program.

23. The computer program product as described in claim 22 wherein said implementation instruction further comprises:

instructions for determining whether said selected program option is possible to implement and;

instructions for sending a decline message to the location to the submission when a selected option is not possible to implement on the identified and retrieved software program.

24. The computer program product as described in claim 22 wherein said implementation instruction further comprises instructions for creating a computing environment for executing the identified and selected software program.

25. The computer program product as described in claim 24 wherein said computer environment creating instruction further comprises instructions for determining the computer hardware and software necessary to implement the selected option on the retrieved software program.

26. The computer program product as described in claim 25 wherein said implementation instructions occur in the database controller.

27. The computer program product as described in claim 25 wherein said implementation instructions occur in a computing environment located at the software program request submission location.

28. A system for maintaining, accessing and executing computer software programs and including a mechanism for storing computer software programs, mechanisms for accessing and executing the stored computer software programs and mechanisms for transmitting and receiving messages over a computer network, said system further comprising:

- a computer connectable to a distributed computing environment;

- a database of stored computer software programs, said programs comprising a collection of programs spanning a chronological range from the 1950's to the present, each said program capable of being executed in said computer;

- a central processor;

- a set of operating system programs to enable the execution of programs stored in said database; and

- a set of emulation and simulation programs for use in the execution of programs stored in said database.